\$	00000000 00000000 00000000	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR		33333333 333333333 3333333333	222222222
\$\$\$ \$\$\$ \$\$\$ \$\$\$	000 000 000 000 000	RRR RRR RRR RRR	<u> </u>	333 333 333 333	222 222 222 222 222
555	000 000	RRR RRR	<u> </u>	333	222
\$\$\$ \$\$\$\$\$\$\$\$\$ \$\$\$\$\$\$\$\$\$	000 000 000 000 000 000	RRR RRRRRRRRRRRR RRRRRRRRRRRRR	111 111 111	333	222
\$\$\$\$\$\$\$\$\$ \$\$\$ \$\$\$	000 000 000 000 000	RRRRRRRRRRRR RRR RRR RRR RRR	<u> </u>	333	222
SSS	000 000	RRR RRR RRR RRR	<u> </u>	333 333	222
\$\$\$ \$\$\$	000 000	RRR RRR RRR RRR RRR RRR	111 111 111	333 33333333	222
\$	00000000	RRR RRR	†††	33333333 333333333	2222222222222222

Pse

_\$2

SOR

SOR

SOR

SOR

_LI

CDD

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

Version: 'V04-000'

TITLE: CDDMAC

CDD Macro Require File

FACILITY: Common Data Dictionary

ABSTRACT:

1 *

This require file contains the CDD Macros used to access the CDD User Interface.

ENVIRONMENT:

AUTHOR: Jeff East and Kenneth J. Marchilena.

21-Oct-80

MODIFIED BY:

P.D.Gilbert 31-Jul-1981

fixed bug with \$CDD\$CREATE_HISTORY with no descriptor parameter

\$R(parm)

This macro checks to make certain a required parameter is present.

MACRO

\$R(parm) =
%IF %NULL (parm) %THEN
%WARN ('Required parameter'

ZQUOTE ZQUOTE ZQUOTE ZQUOTE ZQUOTE ZQUOTE parm, ' missing.')

```
16-SEP-1984 17:02:19.90 Page 2
CDDMAC.R32:1
         XELSE
         %FI
    %:
KEYWORDMACRO
         status.wlc.v = CDD$CLEAR_CELL (context.rlu.r, list.rlu.r,
                          cell.rwu.v);
    $CDD$CLEAR_CELL (context, list, cell) =
         BEGIN
             EXTERNAL ROUTINE
                  CDD$CLEAR_CELL
                                              : FORTRAN;
              CDD$CLEAR_CELL (%EXPAND $R(context), %EXPAND $R(list),
                  %EXPAND $R(cell))
         END
    %,
         status.wlc.v = CDD$CREATE_ACL_ENTRY (context.rlu.r, [path.rt.dx]
                  [node.rlu.r] , position.rwu.v, [grant.rlu.v] , [deny.rlu.v] ,
[banish.rlu.v] , [password.rt.dx] ,
[terminal.rt.dx] , [uic.rt.dx] , [username.rt.dx]);
    $CDD$CREATE_ACL_ENTRY (context, path, node, position, grant=0, deny=0,
    banish=0, password=0, terminal=0, uic=0, username=0) = BEGIN
             EXTERNAL ROUTINE
                  CDD$CREATE_ACL_ENTRY
                                            : FORTRAN;
             CDD$CREATE_ACL_ENTRY (%EXPAND $R(context) %IF %NULL (path) %THEN %IF %NULL (node) %THEN
                            %ERROR ('Either path or node must be specified')
                      XELSÉ 0. 0
                           . 0. node
                       %FI
                  XELSE
                       %IF %NULL (node) %THEN
                      XELSÉ path, 0
                      %FI , path, node
                  %FI
                  , %EXPAND $R(position), grant, deny, banish, password, terminal, uic,
         END
```

CDD

```
16-SEP-1984 17:02:19.90 Page 3
CDDMAC.R32:1
    %.
       $CDD$CREATE_DIR (context, path, node, protocol, options, location) =
        BEGIN
            EXTERNAL ROUTINE
                 CDD$CREATE_DIR
                                           : FORTRAN:
             CDD$CREATE_DIR (%EXPAND $R(context), %EXPAND $R(path)
                 XIF XNULL(node) XTHEN
XIF XNULL(protocol) XTHEN
XIF XNULL (options) XTHEN
XIF XNULL (location) XTHEN
                                  . 0, 0, 0, location)
                          XELSE
                              0. 0. options
%IF %NULL (location) %THEN
                              XELSE.
                              %FI . location)
                          %FI
                     %ELSE
                          21F XNULL (options) XTHEN
                              %IF %NULL (location) %THEN
                              XELSE
                                  , O, location)
                          XELSE.
                              options
%IF %NULL (location) %THEN
                              XELSE
                              %FI . location)
                         XFI
                 XELSE XFI
                       node
                     XIF XNULL (protocol) XTHEN
XIF XNULL (options) XTHEN
XIF XNULL (location) XTHEN
                              XELSE
                                   . O. O. location)
```

CDD

```
XELSE XFI
                       10. options
LIF XNULL (location) %THEN
                       XELSE
                      %FI , location)
              XELSE XFI
                   protocol
%IF %NULL (options) %THEN
%IF %NULL (location) %THEN
                      %FI , 0, location)
                   XELSE.
                       options XIF XNULL (location) XTHEN
                       XELSE
                      %FI , location)
          XFI XFI
   END
X.
   $CDD$CREATE_ENTITY_ATT (context, entity, attribute, location) =
    BEGIN
       EXTERNAL ROUTINE
           CDD$CREATE_ENTITY_ATT : FORTRAN;
       CDD$CREATE_ENTITY_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(location))
    END
%.
    $CDD$CREATE_ENTITY_LIST_ATT (context, entity, attribute, list_size,
location) =
```

```
16-SEP-1984 17:02:19.90 Page 5
CDDMAC.R32:1
        BEGIN
            EXTERNAL ROUTINE
                 CDD$CREATE_ENTITY_LIST_ATT
                                                   : FORTRAN:
            CDD$CREATE_ENTITY_LIST_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(location))
        END
    %.
       $CDD$CREATE_FORWARD (context, path, node=0, file, options, location) =
        BEGIN
            EXTERNAL ROUTINE
                 CDD$CREATE_FORWARD
                                      : FORTRAN;
            CDD$CREATE_FORWARD (%EXPAND $R(context), %EXPAND $R(path), node,
                 XEXPAND SR(file)
                 %IF %NULL (options) %THEN
                     %IF %NULL (location) %THEN
                     XELSE
                         . O. location)
                XELSE
                       options
                     %IF %NULL (location) %THEN
                     XELSE.
                        , location)
                     XFI
                XF I
        END
    %.
        status.wlc.v = CDD$CREATE_HISTORY (context.rlu.r, entity.rlu.r, facility.rt.dx, access.rlu.v,program.rt.dx,
                        description.rt.dx);
    $CDD$CREATE_HISTORY (context, entity, facility, access, program,
        description) = BEGIN
            EXTERNAL ROUTINE
                CDD$CREATE_HISTORY
                                          : FORTRAN;
            CDD$CREATE_HISTORY (%EXPAND $R(context), %EXPAND $R(entity),
```

XEXPAND \$R(facility), XEXPAND \$R(access)

CDI

```
16-SEP-1984 17:02:19.90 Page 6
CDDMAC.R32:1
                    %IF %NULL (program) %THEN
                        %IF %NULL (description) %THEN
                        XELSE
                            , 0, description)
                    XELSE
                        %IF %NULL (description) %THEN
                       %ELSÉ program)
                       %FI , program, description)
                    %FI
       END
   %.
       status.wlc.v = CDD$CREATE_NULL_ATT (context.rlu.r, entity.rlu.r,
                       attribute.rlu.v);
   $CDD$CREATE_NULL_ATT (context, entity, attribute) =
       BEGIN
           EXTERNAL ROUTINE
               CDD$CREATE_NULL_ATT
                                        : FORTRAN;
            CDD$CREATE_NULL_ATT (%EXPAND $R(context), %EXPAND $R(entity),
                %EXPAND $R(attribute))
       END
   %.
       status.wlc.v = CDD$CREATE_NUM_ATT (context.rlu.r, entity.rlu.r,
                       attribute.rlu.v, value.rl.v);
   $CDD$CREATE_NUM_ATT (context, entity, attribute, value) =
       BEGIN
           EXTERNAL ROUTINE
               CDD$CREATE_NUM_ATT
                                        : FORTRAN;
```

CDD\$CREATE_NUM_ATT (%EXPAND \$R(context), %EXPAND \$R(entity),

status.wlc.v = CDD\$CREATE_STRING_ATT (context.rlu.r, entity.rlu.r,

attribute.rlu.v, value.rt.dx , [value_size.rwu.v]);

%EXPAND \$R(attribute), %EXPAND \$R(value))

END

%.

CDI

```
16-SEP-1984 17:02:19.90 Page 7
CDDMAC.R32:1
    $CDD$CREATE_STRING_ATT (context, entity, attribute, value, value_size) =
        BEGIN
            EXTERNAL ROUTINE CDDSCREATE_STRING_ATT
                                                     : FORTRAN;
             CDD$CREATE_STRING_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(value) %IF %NULL (value_size) %THEN
                      , value_size)
        END
    %.
        $CDD$CREATE_STRING_LIST_ATT (context, entity, attribute, list_size,
location) =
        BEGIN
             EXTERNAL ROUTINE
                 CDD$CREATE_STRING_LIST_ATT
                                                     : FORTRAN;
            CDD$CREATE_STRING_LIST_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(location))
        END
    %.
       status.wlc.v = CDD$CREATE_TERM (context.rlu.r, path.rt.dx, [node.rlu.r],
              protocol.rt.dx, options.rlu.v, location.wlu.r , [prior.rt.dx]);
    $CDD$CREATE_TERM (context, path, node=0, protocol, options=0,
         location, prior) =
        BEGIN
             EXTERNAL ROUTINE
                 CDD$CREATE_TERM
                                            : FORTRAN;
```

CDD\$CREATE_TERM (%EXPAND \$R(context), %EXPAND \$R(path), node, XEXPAND \$R(protocol), options, XEXPAND \$R(location)

%IF %NULL (prior) %THEN

XELSE

END

%FI . prior)

CD

```
16-SEP-1984 17:02:19.90 Page 8
CDDMAC.R32:1
   %.
      $CDD$DELETE_ACL_ENTRY (context, path, node, position) =
      BEGIN
          EXTERNAL ROUTINE
             CDD$DELETE_ACL_ENTRY
                                 : FORTRAN;
          CDDSDELETE ACL_ENTRY (%EXPAND $R(context) %IF %NOLL (path) %THEN
                 %IF %NULL (node) %THEN
                    XERROR ('Either path or node must be specified')
                 XELSÉ 0. 0
                XFI . 0, node
             XELSE
                 %IF %NULL (node) %THEN
                XELSÉ path, 0
                %FI , path, node
              , %EXPAND $R(position))
      END
   %.
      $CDD$DELETE_ATT (context, entity, attribute) =
      BEGIN
          EXTERNAL ROUTINE
             CDDSDELETE_ATT
                                  : FORTRAN;
          CDD$DELETE_ATT (%EXPAND $R(context), %EXPAND $R(entity),
             XEXPAND $R(attribute))
      END
   Z.
      status.wlc.v = CDD$DELETE_NODE (context.rlu.r, [path.rt.dx] ,
           [node.rlu.r] , [options.rlu.v]);
```

CDI

```
16-SEP-1984 17:02:19.90 Page 9
CDDMAC.R32:1
   $CDD$DELETE_NODE (context, path, node, options) =
       BEGIN
           EXTERNAL ROUTINE
               CDD$DELETE_NODE
                                       : FORTRAN;
           CDD$DELETE_NODE (%EXPAND $R(context)
               XIF XNULL (path) XTHEN
XIF XNULL (node) XTHEN
                       %ERROR ('Either path or node must be specified')
                   XFI . 0, node
               XELSE
                   %IF %NULL (node) %THEN
                   XELSÉ path
                   %FI , path, node
               %IF %NULL (options) %THEN
               XELSE.
               %FI , options)
       END
   %.
       status.wlc.v = CDD$fILL_STRING_CELL (context.rlu.r, list.rlu.r,
                      cell.rwu.v, value.rt.dx , [value_size.rwu.v]);
   $CDD$FILL_STRING_CELL (context, list, cell, value, value_size) =
       BEGIN
           EXTERNAL ROUTINE
               CDD$FILL_STRING_CELL
                                       : FORTRAN;
           CDD$FILL_STRING_CELL (%EXPAND $R(context), %EXPAND $R(list), %EXPAND $R(cell), %EXPAND $R(value)
               XIF XNULL (value_size) XTHEN
               %FI . value_size)
       END
   %.
```

UN

```
$CDD$FIND_NODE (context, path, node, location, protocol, protocol_size) =
    BEGIN
         EXTERNAL ROUTINE
             CDDSF IND_NODE
                                         : FORTRAN;
         CDD$FIND_NODE (%EXPAND $R(context)
             %IF %NULL(path) %THEN
%IF %NULL (node) %THEN
%ERROR ('Either path or node must be specified')
                         0.0
                 XFI . 0. node
             %ELSE
                  %IF %NULL (node) %THEN
                 XELSE path, 0
                  %FI , path, node
             XIF XNULL (location) XTHEN
                  %IF %NULL (protocol) %THEN
%IF %NULL (protocol_size) %THEN
                           XERROR ('Protocol-size cannot be used without protocol')
                      XF I
                  XELSE
                      10. protocol XIF XNULL (protocol_size) XTHEN
                       XELSE
                      %FI , protocol_size)
                  %FI
             XELSE
                     location
                  XIF XNULL (protocol) XTHEN
XIF XNULL (protocol_size) XTHEN
                       XELSE
                           XERROR ('Protocol-size cannot be used without protocol')
                  XELSE XFI
                       #IF #NULL (protocol_size) #THEN
                       XELSE
                      %FI , protocol_size)
            XFI XFI
    END
```

```
16-SEP-1984 17:02:19.90 Page 11
CDDMAC.R32:1
   %.
       $CDD$FORMAT_ACL_ENTRY (context, node, position, string, string_size) =
       BEGIN
           EXTERNAL ROUTINE
              CDD$FORMAT_ACL_ENTRY
                                     : FORTRAN;
           CDD$FORMAT_ACL_ENTRY (%EXPAND $R(context), %EXPAND $R(node), %EXPAND $R(position), %EXPAND $R(string)
               %IF %NULL (string_size) %THEN
               XELSE
              %FI , string_size)
       END
   %.
       $CDD$GET_ACCESS_RIGHTS (context, path, node, rights) = BEGIN
           EXTERNAL ROUTINE
               CDD$GET_ACCESS_RIGHTS : FORTRAN;
           CDD$GET_ACCESS_RIGHTS (%EXPAND $R(context) 
%IF %NULL (path) %THEN 
%IF %NULL (node) %THEN
                      XERROR ('Either path or node must be specified')
                        0.0
                  XELSÉ
                  XFI . 0. node
               XELSE
                  XIF XNULL (node) XTHEN
                  XELSÉ path, 0
                  %FI , path, node
               , %EXPAND $R(rights))
       END
```

```
CDDMAC.R32;1
```

```
$CDD$GET_ACL_ENTRY (context, node, position, grant, deny,
banish, password, terminal, uic, username) = BEGIN
        EXTERNAL ROUTINE
             CDD$GET_ACL_ENTRY
                                        : FORTRAN;
        CDD$GET_ACL_ENTRY (%EXPAND $R(context), %EXPAND $R(node), %EXPAND $R(position), %EXPAND $R(grant), %EXPAND $R(deny), %EXPAND $R(banish), %EXPAND $R(password), %EXPAND $R(terminal), %EXPAND $R(uic), %EXPAND $R(username))
        END
    $CDD$GET_ATT (context, entity, attribute, type, location, value, value_size) =
    BEGIN
        EXTERNAL ROUTINE
             CDD$GET_ATT
                                        : FORTRAN:
         CDD$GET_ATT (%EXPAND $R(context), %EXPAND $R(entity),
             XEXPAND $R(attribute), XEXPAND $R(type)
XIF XNULL (location) XTHEN
                  XIF XNULL (value) XTHEN
                      %IF %NULL (value_size) %THEN
                      %FI . 0, 0, value_size)
                 XELSE
                      XIF XNULL (value_size) XTHEN
                     XELSÉ O, value)
                     %FI ,0, value, value_size)
                 XF I
             XELSE
                 XIF XNULL (value) XTHEN
XIF XNULL (value_size) XTHEN
                     XELSÉ location)
```

```
16-SEP-1984 17:02:19.90 Page 13
CDDMAC.R32:1
                        %FI , location, 0, value_size)
                        %IF %NULL (value_size) %THEN
                        %ELSÉ location, value)
                       %FI , location, value, value_size)
               XFI XFI
        END
        status.wlc.v = CDD$GET_ATTS (context.rlu.r, location.rlu.r,
                       list.ra.v);
   $CDD$GET_ATTS (context, location, list) =
        BEGIN
            EXTERNAL ROUTINE
                CDD$GET_ATTS
                                        : FORTRAN;
            CDD$GET_ATTS (%EXPAND $R(context), %EXPAND $R(location),
                XEXPAND SR(List))
        END
   %.
        status.wlc.v = CDD$GET_ENTITY_ATT (context.rlu.r, entity.rlu.r,
                       attribute.rlu.v, location.wlu.r);
   $CDD$GET_ENTITY_ATT (context, entity, attribute, location) = BEGIN
            EXTERNAL ROUTINE
                CDD$GET_ENTITY_ATT
                                        : FORTRAN;
            CDD$GET_ENTITY_ATT (%EXPAND $R(context), %EXPAND $R(entity),
                %EXPAND $R(attribute), %EXPAND $R(location))
        END
        status.wlc.v = CDD$GET_ENTITY_CELL (context.rlu.r, list.rlu.r,
                       cell.rwu.v, location.wlu.r);
   $CDD$GET_ENTITY_CELL (context, list, cell, location) = BEGIN
```

LI

LI

LI

```
16-SEP-1984 17:02:19.90 Page 14
CDDMAC.R32:1
             EXTERNAL ROUTINE
                 CDD$GET_ENTITY_CELL
                                             : FORTRAN;
             CDD$GET_ENTITY_CELL (%EXPAND $R(context), %EXPAND $R(list), %EXPAND $R(location))
    %.
        $CDD$GET_ENTITY_LIST_ATT (context, entity, attribute, location,
    list_size) =
BEGIN
             EXTERNAL ROUTINE
                  CDD$GET_ENTITY_LIST_ATT : FORTRAN;
             CDD$GET_ENTITY_LIST_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(location) %IF %NULL (list_size) %THEN
                  XELSE.
                      , list_size)
        END
    %.
        status.wlc.v = CDD$GET_NEXT_ATT (context.rlu.r, entity.rlu.r,
                          attribute.wlu.r, type.wlu.r, [location.wlg.r], [string.rt.dx], [value_size.rwu.r]);
    $CDD$GET_NEXT_ATT (context, entity, attribute, type, location, string,
                          value_size) =
        BEGIN
             EXTERNAL ROUTINE
                  CDD$GET_NEXT_ATT
                                             : FORTRAN;
             CDD$GET_NEXT_ATT (%EXPAND $R(context), %EXPAND $R(entity),
                  XEXPAND SR(attribute), XEXPAND SR(type)
XIF XNULL (location) XTHEN
                      %IF %NULL (string) %THEN
                           %IF %NULL (value_size) %THEN
                               . O. O. value_size)
                      XELSE
```

!-

LI

LI

```
16-SEP-1984 17:02:19.90 Page 15
CDDMAC.R32:1
                          %IF %NULL (value_size) %THEN
                          XELSÉ 0, string)
                         %FI ,0, string, value_size)
                XELSE XIF
                      XF I
                          %NULL (string) %THEN %IF %NULL (yalue_size) %THEN
                         %ELSÉ location)
                          %FI , location, 0, value_size)
                          %IF %NULL (value_size) %THEN
                          XELSÉ location, String)
                         %FI . location, string, value_size)
                XFI XFI
        END
        status.wlc.v = CDD$GET_NULL_ATT (context.rlu.r, entity.rlu.r,
                         attribute.rlu.v);
    $CDD$GET_NULL_ATT (context, entity, attribute) =
        BEGIN
            EXTERNAL ROUTINE
                 CDD$GET_NULL_ATT
                                            : FORTRAN;
            CDD$GET_NULL_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute))
        END
   %.
        status.wlc.v = CDD$GET_NUM_ATT (context.rlu.r, entity.rlu.r,
                         attribute.rlu.v, value.wl.r);
    $CDD$GET_NUM_ATT (context, entity, attribute, value) =
        BEGIN
            EXTERNAL ROUTINE
                                            : FORTRAN;
                 CDD$GET_NUM_ATT
             CDD$GET_NUM_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(value))
        END
```

```
BODEFGHIJKLMNBODEFGHIJKLMNBODEFGHIJK
LMNBCDEFGHIJKLMNBCDEFGHI
```

```
CDDMAC.R32:1
    %.
         status.wlc.v = CDD$GET_STRING_ATT (context.rlu.r, entity.rlu.r,
                          attribute.rlu.v, value.wt.dx , [value_size.wwu.r]);
    $CDD$GET_STRING_ATT (context, entity, attribute, value, value_size) = BEGIN
             EXTERNAL ROUTINE
                  CDD$GET_STRING_ATT
                                             : FORTRAN:
             CDD$GET_STRING_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(value) %IF %NULL (value_size) %THEN
                  XELSE
                      , value_size)
                  %FI
         END
    %.
        status.wlc.v = CDD$GET_STRING_CELL (context.rlu.r, list.rlu.r,
                          cell.rwu.v, value.wt.dx , [value_size.wwu.r]);
    $CDD$GET_STRING_CELL (context, list, cell, value, value_size) =
BEGIN
             EXTERNAL ROUTINE
                  CDD$GET_STRING_CELL
                                             : FORTRAN;
             CDD$GET_STRING_CELL (%EXPAND $R(context), %EXPAND $R(list), %EXPAND $R(cell), %EXPAND $R(value)
                  %IF %NULL (value_size) %THEN
                  XELSE
                      , value_size)
        END
    X.
        status.wlc.v = CDD$GET_STRING_LIST_ATT (context.rlu.r, entity.rlu.r,
                          attribute.rlu.v, location.wlu.r ,
                          [list_size.wwu.r]);
    $CDD$GET_STRING_LIST_ATT (context, entity, attribute, location,
         list_size) =
```

```
16-SEP-1984 17:02:19.90 Page 17
CDDMAC.R32:1
         BEGIN
             EXTERNAL ROUTINE
                  CDD$GET_STRING_LIST_ATT : FORTRAN;
             CDD$GET_STRING_LIST_ATT (%EXPAND $R(context), %EXPAND $R(entity), %EXPAND $R(attribute), %EXPAND $R(location) %IF %NULL (list_size) %THEN
                  XELSE
                 %FI , list_size)
        END
       status.wlc.v = CDD$LOCK_NODE (context.rlu.r, [path.rt.dx] ,
               [node.rlu.r] , location.wlu.r , [protocol.wt.dx] ,
               [protocol-size.wwu.r]);
    $CDD$LOCK_NODE (context, path, node, location, protocol, protocol_size) =
        BEGIN
             EXTERNAL ROUTINE
                  CDD$LOCK_NODE
                                             : FORTRAN:
             CDD$LOCK_NODE (%EXPAND $R(context)
                  %IF %NULL(path) %THEN
                      %IF %NULL (node) %THEN 
%ERROR ('Either path or node must be specified')
                      XELSÉ 0. 0
                      %FI . 0, node
                  XELSE
                      %IF %NULL (node) %THEN
                      XELSÉ path, 0
                      %FI , path, node
                  %FI
                    %EXPAND $R(location)
                  XIF XNULL (protocol) XTHEN
XIF XNULL (protocol_size) XTHEN
                           XERROR ('Protocol-size cannot be used without protocol')
                      XF I
                  XELSE
                      #IF #NULL (protocol_size) #THEN
                       XELSE
                           , protocol_size)
```

```
16-SEP-1984 17:02:19.90 Page 18
CDDMAC.R32:1
                     XF I
                 XF I
        END
    %.
       [protocol-size.wwu.r]);
    $CDD$NEXT_NODE (context, node, name, name_size, location, protocol,
protocol_size) =
    BEGIN
            EXTERNAL ROUTINE
                 CDD$NEXT_NODE
                                           : FORTRAN:
            CDD$NEXT_NODF (%EXPAND $R(context), %EXPAND $R(node),
                 %EXPAND $R(name)
%IF %NULL (name_size) %THEN
%IF %NULL (Tocation) %THEN
                          %IF %NULL (protocol) %THEN
%IF %NULL (protocol_size) %THEN
                                  XERROR ('Protocol-size cannot be used without protocol')
                              %FI
                         XELSE
                              21F %NULL (protocol_size) %THEN
                              XELSE.
                             %FI , protocol_size)
                          XF I
                     XELSE
                            O, location
                         %IF %NULL (protocol) %THEN
%IF %NULL (protocol_size) %THEN
                              XELSE
                                  XERROR ('Protocol-size cannot be used without protocol')
                              XFI
                          XELSE.
                              #IF #NULL (protocol_size) #THEN
                              XELSE
                             %FI , protocol_size)
                         XF I
```

XFI

END

BEGIN

X.

```
XELSE
                   name_size

%IF %NULL (location) %THEN

%IF %NULL (protocol) %THEN

%IF %NULL (protocol_size) %THEN
                                  XERROR ('Protocol-size cannot be used without protocol')
                             XFI
                        XEL SE
                             1. protocol
LIF XNULL (protocol_size) %THEN
                             XELSE
                             %FI , protocol_size)
                        XF I
                   XELSE.
                           location
                        %IF %NULL (protocol) %THEN
%IF %NULL (protocol_size) %THEN
                             XELSE
                                  XERROR ('Protocol-size cannot be used without protocol')
                             XFI
                        XELSE.
                             #IF #NULL (protocol_size) #THEN
            %FI %FI %FI protocol_size)
   status.wlc.v = CDD$RENAME_NODE (context.rlu.r, [path.rt.dx] ,
           [node.rlu.r] , name.rt.dx);
$CDD$RENAME_NODE (context, path, node, name) =
         EXTERNAL ROUTINE
               CDD$RENAME_NODE
                                            : FORTRAN;
          CDD$RENAME_NODE (%EXPAND $R(context) %IF %NULL(path) %THEN %IF %NULL (node) %THEN
                        XERROR ('Either path or node must be specified')
```

CDDMAC.R32:1

, %EXPAND \$R(name)) END %. \$CDD\$RLSE_LOCKS (context, path, node, options) =
BEGIN EXTERNAL ROUTINE CDD\$RLSE_LOCKS : FORTRAN; CDD\$RLSE_LOCKS (%EXPAND \$R(context)
%IF %NULL(path) %THEN
%IF %NULL(node) %THEN
%IF %NULL(options) %THEN . 0, 0, options) %FI XELSE. 0, node %IF %NULL(options) %THEN XELSE. %FI , options) XELSE XFI IF INULL (node) ITHEN %IF %NULL(options) %THEN XFI . 0. options) XELSE XIF XNULL (options) XTHEN

XELSE.

XELSÉ 0. 0

XELSÉ path, 0

%FI , path, node

XFI
XELSE
XIF XNULL (node) XTHEN

```
16-SEP-1984 17:02:19.90 Page 21
CDDMAC.R32:1
                %FI %FI , options)
        END
       status_wlc.v = CDD$SET_DEFAULT (context.rlu.r, [path.rt.dx] ,
              [node.rlu.r]);
    $CDD$SET_DEFAULT (context, path, node) = BEGIN
             EXTERNAL ROUTINE
                 CDD$SET_DEFAULT
                                            : FORTRAN;
             CDD$SET_DEFAULT (%EXPAND $R(context)
%IF %NULL(path) %THEN
%IF %NULL (node) %THEN
%ERROR ('Either path or node must be specified')
                     XELSE 0, 0)
                     %FI . 0, node)
                 XELSE
                      %IF %NULL (node) %THEN
                      XELSÉ path, 0)
                     %FI , path, node)
                 %FI
        END
    %.
       status.wlc.v = CDD$SIGN_IN (context.wlu.r , [default_dir.rt.dx]);
    $CDD$SIGN_IN (context, default_dir) =
BEGIN
             EXTERNAL ROUTINE
                  CDD$SIGN_IN
                                             : FORTRAN;
             CDD$SIGN_IN (%EXPAND $R(context)
                  XIF XNULL (default_dir) XTHEN
                  XELSE
                 %FI . default_dir)
        END
    X.
```

0362 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

